VLSI Implementation of AES Block Cipher Based Data Hiding in Image Processing

Kumarganesh S.*[1], Socratees P.[2], Rajamanickam G.[3]

[1]Professor, Department of ECE, Knowledge Institute of Technology, Salem, Tamil Nadu, India.
[2]PG Scholar, Department of ECE, Knowledge Institute of Technology, Salem, Tamil Nadu, India.
[3]Assistant Professor, Department of ECE, Knowledge Institute of Technology, Salem, Tamil Nadu, India.
Emails:saikgss@gmail.com; socratees1046@gmail.com; grece@kiot.ac.in

Abstract

A common symmetric key block cypher for protecting electronic data is called the Advanced Encryption Standard (AES). The implementation of the AES algorithm using Very-Large-Scale Integration (VLSI) is examined in this study. Cryptographic algorithms can be realised in hardware thanks to VLSI, which has advantages over software implementations in terms of increased processing speed and security. In this work, an AES block cypher core designed and implemented with VLSI techniques is described. We investigate various architectural designs to maximise performance indicators such as area, power consumption, and throughput. The trade-offs between different implementation options and design concerns for important components such as S-boxes will be covered. To illustrate the performance and usefulness of the implemented AES core, simulation results will be given. This paper examines the VLSI implementation of the Advanced Encryption Standard (AES), with a focus on image processing applications, even though it is essential for general data security. Effective on-chip encryption and decryption can be included into image processing systems by implementing the AES algorithm into specialised hardware circuits. Benefits of this VLSI design include enhanced performance over software-based solutions on resource-constrained image processing devices, the ability to encrypt images in real-time for secure transmission or storage, and the potential for reduced power usage for battery-powered applications.

Keywords: Advanced Encryption Standard (AES); Very-Large-Scale Integration (VLSI) Algorithm; Block cipher; Data hiding; S-boxes

1. Introduction

One popular and very safe block cypher is called Advanced Encryption Standard, or AES. It is a fundamental technology in contemporary cryptography, protecting private information both in transit and in storage. Significant progress in data concealing and cryptography has been fueled by the constant need for secure communication and data storage. By putting forth a VLSI implementation of an AES block-cipher-based data concealing technique for image processing applications, this study investigates the integration of these two fields. Important features of it are:

- Block Cypher: AES uses fixed-size blocks (usually 128 bits) to encrypt data, in contrast to stream cyphers that encrypt data one bit at a time. It uses a secret key to separately encrypt each block.
- Symmetric vs. Asymmetric: AES is a symmetric cipher, meaning the same key is used for both encryption and decryption. This offers simplicity but requires secure key exchange beforehand.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

- Key Length and Rounds: AES encryption is stronger as keys get longer. Three key lengths are supported: 128 bits, 192 bits, and 256 bits. Along with key length, the number of encryption rounds, or iterations, vary as well; more rounds provide more robust protection.

✓ 128-bit key: 10 rounds
✓ 192-bit key: 12 rounds
✓ 256-bit key: 14 rounds

Numerous cryptography systems, such as the DES algorithm, offer some degree of security; however, because it only has 56 key lengths, it is susceptible to easy hacking. To improve reliability, the National Institute of Standards and Technology (NIST) developed 15 very secure algorithms. For secure data transfer, the Rijndael algorithm took the place of the DES in October 2000 and became known as the Advanced Encryption Standard (AES). Numerous fields, including cellular networks, web servers, mobile networks, smart cards, etc., use the AES standard. AES is used by digital video recorders for both picture transmission and reception.

The AES algorithm is thought to be more dependable, and its implementation at the hardware and software levels is now simpler and more effective. It is more dependable to implement in hardware such as Field Programmable Gate arrays (FPGA). FPGA aids in design optimisation and prototyping for various design requirements. Numerous projects with varying degrees of optimisation have been completed, implemented using FPGA, and customised to create an ASIC device.

A. AES Block Cipher Process

AES encrypts data through a series of intricate steps, broadly categorized as:

- State Array: The message to be encrypted (plaintext) is divided into 16 bytes (128 bits) and arranged into a 4x4 matrix called the state array.
- Key expansion: To create round keys that are used throughout the encryption process, the secret key is subjected to a different transformation.
- Round Function: The round function, which is the foundation of AES, loops through the key schedule and state array. It consists of four fundamental actions that are repeated:

➢ SubBytes: Using a preset S-box, byte-level replacement that ensures dispersion, which disperses the impact of a single bit change throughout the ciphertext.
➢ ShiftRows: The state array's rows are cyclically shifted by a predetermined number of positions.
➢ Mixing: the state array's columns by a particular mathematical procedure can improve diffusion even more.
➢ AddRoundKey: Using a round key generated from the enlarged key schedule, XORing the state array incorporates the key into the encryption process.

- Final Round: The final round differs slightly, omitting the MixColumns step.
- Ciphertext Output: The resulting state array after all rounds is the encrypted message (ciphertext).

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

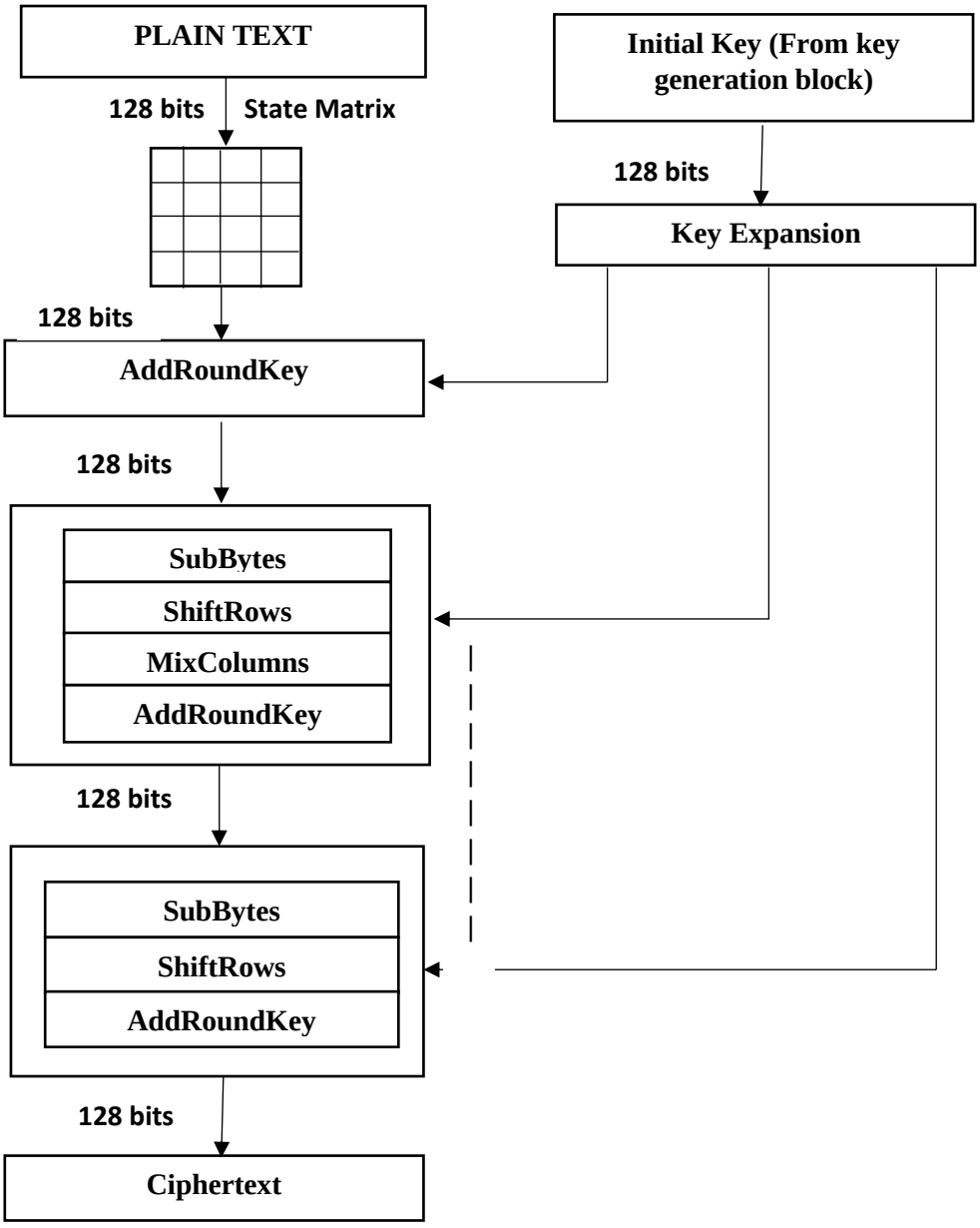

Figure 1: Block diagram for AES process

Block Diagram for the AES Process as Illustrated in Figure 1. Decryption essentially reverses the encryption process, using the same key but applying the round function steps in reverse order with specific round keys. This transforms the ciphertext back to the original plaintext.

B. VLSI Implementation for Hardware Acceleration

The design and production of integrated circuits containing millions of transistors on a single chip is known as Very-Large-Scale Integration, or VLSI. Using VLSI hardware to implement the data concealing method provides:

- High Throughput: When compared to software-based solutions, hardware implementations can achieve far quicker processing speeds.
- Low Power Consumption: VLSI designs are appropriate for devices with limited resources since they can be optimised for power efficiency.

C. Project Objectives

The goal of this research is to create a data hiding method for image processing applications that is both secure and effective. Among the specific goals are:

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

- Creating a method for cipher-based data concealing in images using an AES block cypher.
- Creating a VLSI architecture to implement the suggested data concealing strategy in real time.
- Assessing the VLSI design's performance in terms of throughput, power consumption, and hardware resource usage.

2. Related Work

AES (Advanced Encryption Standard) itself is an algorithm, but there are various technologies that enable its implementation for secure data encryption. Here's a breakdown of existing technologies for AES:

Zhi Li et al. [1] described AES (Advanced Encryption Standard) techniques were improved for system architecture and algorithm design, and hardware was integrated to speed up technological advancements. According to the experimental results, the AES algorithm lowered the encryption time by an average of 40% when compared to the DES algorithm for the same quantity of data. This demonstrated the necessity of strengthening key management even further. To increase the security performance of the accounting data encryption processing system using the DES algorithm, it must set up a mechanism for data backup and recovery, reinforce security awareness raising, and provide security training. The confidentiality and integrity of accounting data would be guaranteed by these advancements, which would give businesses access to more dependable and efficient accounting data protection solutions. Haider K. Hoomod et al. [2] proposed an algorithm that can replace traditional encryption algorithms in the world, which can also be selected by the (NIST) as an advanced coding standard. Here, he showed the modification to improvement of the Twofish algorithm, which is used to make the algorithm better than the memory consumption with the S-boxes, and better security, through the hybridization between processes that operate on the SALSA20 algorithm with the techniques in the S-BOXES of the Twofish algorithm and replace the key-schedule in s-boxes with the equation in system, that will have caused difficulty of breaking the code generated by the s -boxes exists in this algorithm.

Malli Mahendra et al. [3] ensured that the information with limiting the spillage and loss of information, by fostering the privacy, integrity of information move through the cloud and improve the trust pace of the users. To give security in the cloud Novel Data Authentication is used. To guarantee security a critical plan is executed that is against the cipher attack. A total sample size of 20 is performed on two gatherings to accomplish the better encryption time. Farah Jihan Aufa et al. [4] asymmetric keys or public key cryptography are frequently employed in information and communication systems data security. Because of its simplicity, the RSA algorithm (Rivest, Shamir, and Adleman) is among the most well-liked and frequently applied public key cryptography techniques. The encryption and decryption processes are the two primary uses of RSA. The Digital Signature Standard's (DSS) Digital Signature Algorithm (DSA) is the digital signature algorithm that is used. The public key cryptography scheme also includes DSA. Digital signature creation and validity verification are the two primary uses for DSA. The purpose of this study is to determine which bits are more useful by comparing the computing times of RSA with various bits. Then combine both RSA and DSA algorithms to improve data security. From the simulation results, the authors choose RSA 1024 for the encryption process and added digital signatures using DSA 512, so the messages sent are not only encrypted but also have digital signatures for the data authentication process.

M. A. El-Mowafy et al. [5] suggested two newly techniques for compressed videos using the cutting-edge H.264/AVC video coding are given. A strong video encryption technique based on chaotic maps and a random key was used to test the first algorithm approach under various attack scenarios. For video frames, a mix of steganography and cryptography based on chaotic maps has been used to accomplish the second algorithm technique. Using MATLAB software, the suggested techniques are applied to luminance component Y of a collection of several YUV video sequences with varying resolutions. Using a variety of performance measures, the simulation results of the suggested algorithms are assessed in comparison to those obtained using cutting-edge methods. It is advised to use the proposed methodologies in real-time applications going forward since they have demonstrated greater resistance to various types of attacks. Vanitha M. et al. [6] offered a productive VLSI design to boost the Advanced Encryption Standard (AES) Algorithm's throughput and security. While the AES algorithm uses look-up tables for subbyte transformations and inverse subbyte transformations, our suggested method makes use of combinational circuit and pipelining techniques to decrease latency and boost performance. This design suggests a new method for implementing the S-box, which controls the AES architecture's speed and power. By employing pseudo-nMOS technology to make

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

the architecture's fundamental parts fully error detectable, this design also boosts system security. After thorough routing, a throughput of 58.8 Gbps was attained for this AES design, which was modeled using Verilog HDL and synthesized using TSMC's 90 nm standard cell library and RTL Compiler. The physical design implementation was carried out using SOC Encounter.

Karim Shahbazi et al. [7] provides a high-secure symmetric cryptography algorithm that is lightweight and can be implemented on field-programmable gate arrays (FPGAs) and 65-nm technology for Internet of Things devices with limited resources. Five primary blocks and an 8-bit datapath are features of the suggested architecture. To store the plain text, keys, and intermediate data, we create two designated register banks: Key-Register and StateRegister. Shift-Rows is integrated into the StateRegister in order to minimize the area. We propose an optimized 8-bit block for Mix-Columns with four internal registers that accept 8-bit and return 8-bit in order to adapt the Mix-Column to an 8-bit datapath. Additionally, the encryption and key expansion phases use a common optimized Sub-Bytes. Gaurav Ramtri et al. [8] suggested use cryptography methods. A text can be encrypted using cryptographic procedures, which only an authorized user can decrypt. We now suggest combining the two such methods, the two fish and RSA algorithms. As an asymmetric cryptographic technique, RSA requires a pair of keys, referred to as public and private keys. The Two Fish Algorithm is a block cipher that has a 256-bit key length. It is claimed to be effective even in situations when the company employs modest processors. It lets implementers balance performance by letting them trade off encryption speed, key setup time, and code size.

David Smekal et al. [9] described the hardware-accelerated implementation of the Twofish encryption algorithm on Field Programmable Gate Array (FPGA) network cards. The encryption core was implemented using the Virtex 7 network card to achieve real-time encryption and decryption. The algorithm was implemented for 128-bit words and 128-bit keys. This article demonstrates that the Twofish encryption core can operate with the maximum clock frequencies of 315 MHz and achieves the throughput of 48 Gbps, which is faster than most currently implemented systems. Venkata Ramaiah Kavuri et al. [10] intended to stop malware attacks, data leaks, theft of intellectual property, hacker control, and insecure access point management. The suggested system concentrates on several cloud technologies, including private, multi, hybrid, and public clouds, in order to offer services to clients while reducing security risks, expenses, and data leakage issues. A vast array of clever and sophisticated devices are made possible by IoT [11]. These gadgets serve both distant and local purposes. These days, data on the internet needs to be secured via encryption techniques. The encryption algorithm known as the Dynamic Multiple Clouds Cryptographic Algorithm (DMCCA) is used. This algorithm combines the Blow Fish and RSA techniques. The efficiency and speed of the DMCCA method are high when compared to the RSA and Blow Fish techniques [12]. This algorithm combines the Blow Fish and RSA techniques. The efficiency and speed of the DMCCA method are high when compared to the RSA and Blow Fish techniques.

3. Proposed methodology

The two main phases of the suggested system are decryption and data concealing. The system receives an original image, a secret message, and an encryption key as inputs during the data concealing phase. The AES block cypher algorithm is used to first encrypt the secret communication. Subsequently, the original image is integrated with the encrypted data. Typically, the Least Significant Bit (LSB) modification is used in this situation. The hidden data is included with minimal impact on the image's visual quality by altering the least significant bits of the pixels. The main goal of the VLSI implementation is to design hardware modules specifically for the LSB modification method and the AES encryption procedure. These modules are made with high throughput, low power consumption, and effective hardware utilisation in mind. The encrypted data is embedded into the image pixels by the LSB modification module [13]. To provide an extra degree of protection, the data is first encrypted using the Advanced Encryption Standard (AES) block cypher. Very-Large-Scale Integration (VLSI) hardware is then used to speed up the entire process, providing benefits including quicker processing, less power consumption, and maybe higher throughput in comparison to software-based implementations. The AES Encryption Block Diagram as Depicted in Figure 2.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

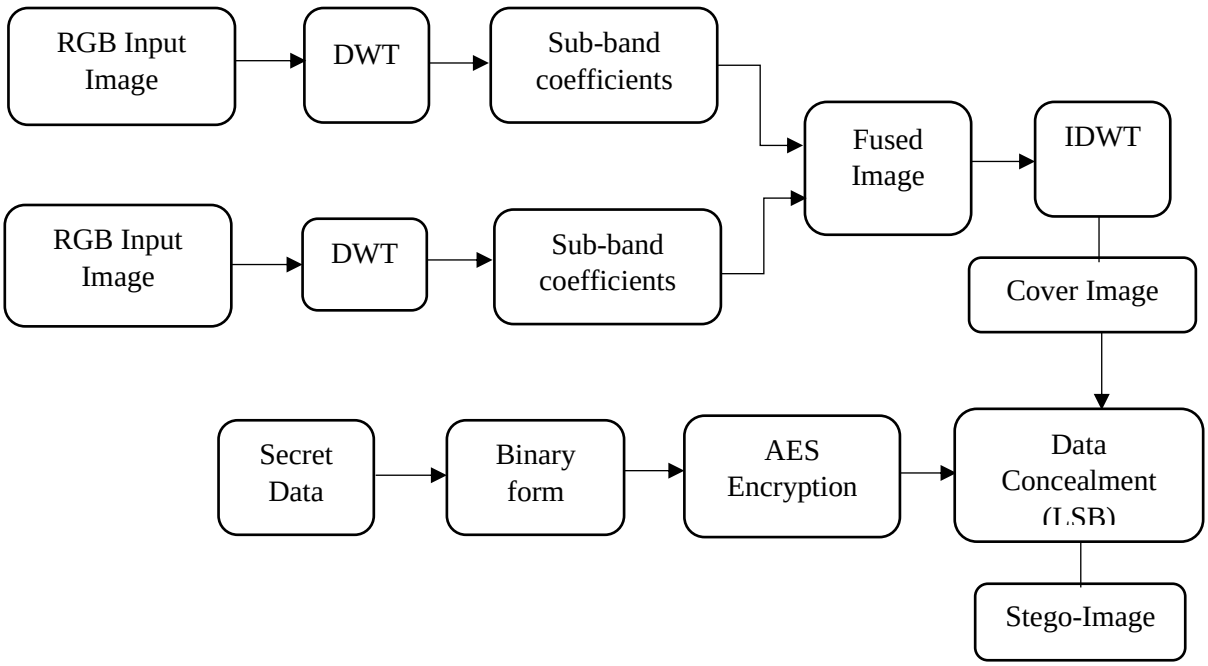

Figure 2: Block Diagram for AES Encryption

- **RGB Input Image:** This is the source image that will be utilised to conceal the confidential information.
- **DWT:** Discrete Wavelet Transform is represented by this block. The picture is broken down mathematically into its individual sub-bands. At various frequencies, these sub-bands hold information about the image.
- **Sub-band coefficients:** The coefficients found in this block are those that are acquired by applying the DWT to the input image. The image in the frequency domain is represented by these coefficients.
- **Secret Data:** This is the information you wish to keep hidden within the picture. It can be any kind of data, including text and audio.
- **Binary form:** This block denotes the conversion of secret data into binary form prior to its concealment within the image. The majority of computers use binary (0s and 1s) for data processing and storage.
- **IDWT:** The Inverse Discrete Wavelet Transform is represented by this block. The process of reconstructing the image from its sub-bands involves mathematics.
- **Fused Image:** Following the embedding of the secret data, this is the final image produced. It is produced by utilising the modified sub-bands with the IDWT.
- **Advanced Encryption Standard (AES):** This block describes the widely used encryption algorithm that can be used to jumble sensitive data before concealing it. This strengthens the steganographic process's security further.
- **Data Concealment (LSB):** This block describes how the secret data is concealed within the picture. The way LSB substitution operates is by swapping out the secret data bits for the least significant bits in each pixel of the image.
- **Cover Image:** After the secret data has been concealed inside it, this is the last image that is produced. Although the cover image seems normal, the concealed data is actually contained in it.

For increased security and effectiveness, the system uses hardware implementation in conjunction with data concealing and picture processing. It uses a method similar to Least Significant Bit (LSB) replacement to conceal secret data (text, pictures, or other digital information) in an image [14]. To provide an extra degree of protection, the data is first encrypted using the Advanced Encryption Standard (AES) block cypher. Diagrammatic Representation of AES Decryption in Figure 3.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

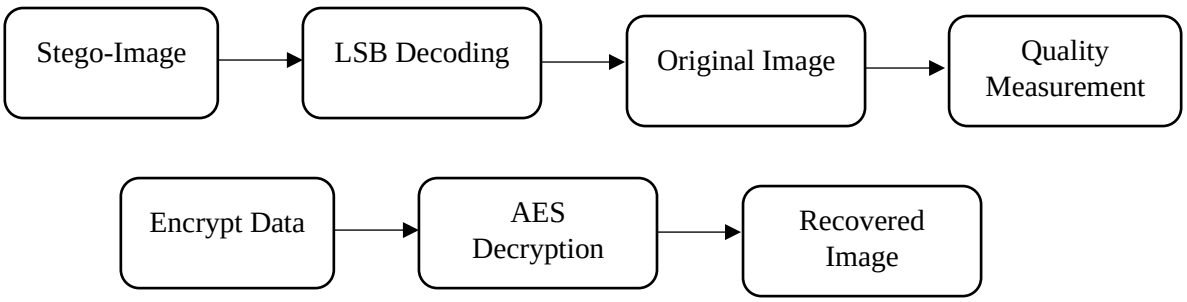

Figure 3: Block Diagram for AES Decryption

Very-Large-Scale Integration (VLSI) hardware is then used to speed up the entire process, less power consumption, and maybe higher throughput in comparison to software-based implementations. Performance and security benefits abound, but there are drawbacks as well, such as complicated hardware designs, effects on image quality, and key management.

A. AES Algorithm

While AES itself is a well-established and secure system, there's always room for improvement depending on the specific application. Here are some potential areas for proposing a system that builds upon AES

a. Pre-processing and Post-processing

• **Pre-processing:** Before feeding the data into AES, you could suggest a system that includes pre-processing processes [15]. This could entail methods such as: Integrity checking involves using a different key to add the message authentication code (MAC) to the plaintext. Schemes for Padding: Prior to encryption, the plaintext is padded to a particular block size.

• **Post-processing:** Following AES decryption, you may suggest the following post-processing actions: Authentication: During pre-processing, confirming the message integrity using the MAC that was previously connected. Depadding: To recover the original plaintext, remove the padding bits that were inserted during pre-processing.

b. Key Management

A safe key exchange mechanism is essential to AES. Key Agreement Protocols: These protocols use key exchange to safely create a shared secret key between parties that communicate with one other without sending the key over an unsecured channel. Key Hierarchies: Putting forward a plan for a system that makes use of master keys and derived keys for various uses.

c. Mode of Operation

Despite being a block cypher, AES is frequently used with larger amounts of data in practical applications. Popular modes consist of: A chain dependency is created when the current round key and the prior ciphertext block are used to encrypt the CBC block [16]. A pseudo-random stream is created by encrypting a counter value with the key and XORing it with the plaintext combines message authentication and counter mode to provide both confidentiality and integrity. The Advanced Encryption Standard (AES) process is visually represented in Figure 4 through a comprehensive block diagram.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

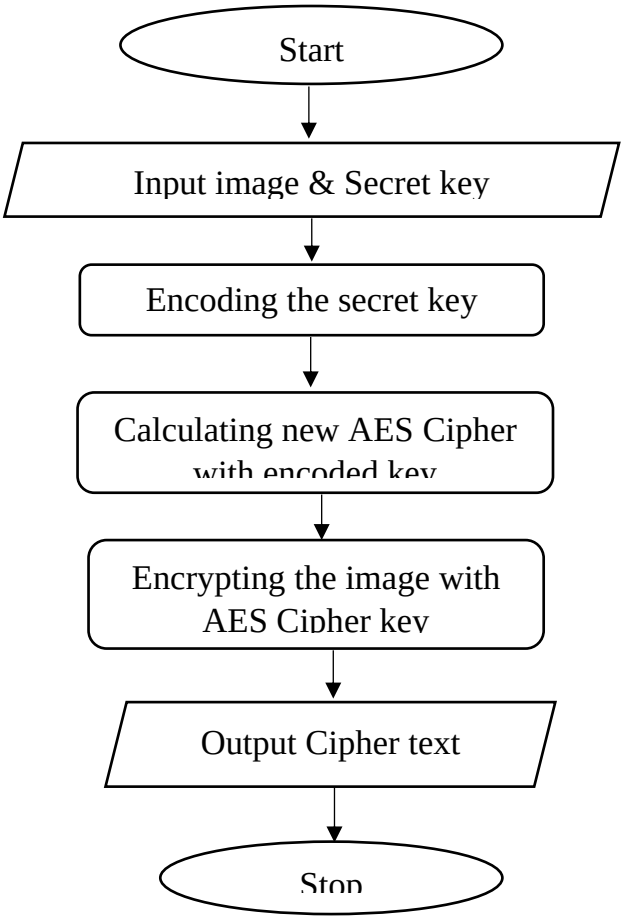

Figure 4: AES Flowchart

d. Image Steganalysis and Security

To determine how imperceptible the embedded data is, a steganalysis of the stego-image (picture containing hidden data) should be performed. Finding the fewest possible visual artefacts or statistical differences from the original image is part of this process.

B. AES Architecture and its equation

SubBytes and ShiftRows are two order-independent transformations. In reality, the Round Function Module's MixColumns transformation is the matrix multiplied over the finite field GF (2^8). The particular computations are used to show how the SubBytes, ShiftRows, and MixColumns transformations in the round function may be replaced with a technique of looking up a single 256-byte table. This can be stated using the matrix form as follows:

$$: \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} s(a_{0,0}) & s(a_{0,1}) & s(a_{0,2}) & s(a_{0,3}) \\ s(a_{1,0}) & s(a_{1,1}) & s(a_{1,2}) & s(a_{1,3}) \\ s(a_{2,0}) & s(a_{2,1}) & s(a_{2,2}) & s(a_{2,3}) \\ s(a_{3,0}) & s(a_{3,1}) & s(a_{3,2}) & s(a_{3,3}) \end{pmatrix} = \begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} \quad (1)$$

(For $0 \leq c < 4$)

The intermediate ciphertext acquired after the SubBytes and ShiftRows transformations are carried out is represented by the value of the state matrix in the middle of equation (1), which also shows the constant matrix for the MixColumns operation on the left. $S(a_{i,j})$, The replacement value for $a_{i,j}$, which is found by consulting the S-box, is the element of the middle matrix. After completing the MixColumns transformation, the intermediate ciphertext is found by computing Eq. (1), which is represented by the value of the state

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

matrix on the right. By substituting and for finite field multiplication and addition (XOR), respectively, the aforementioned computations can be written as follows:

$$S'_{0,c} = 2.s(a_{0,c}) \oplus 3.s(a_{1,c}) \oplus s(a_{2,c}) \oplus s(a_{3,c})$$
$$S'_{1,c} = s(a_{0,c}) \oplus 2.s(a_{l,c}) \oplus 3.s(a_{2,c}) \oplus s(a_{3,c})$$
$$S'_{2,c} = s(a_{0,c}) \oplus s(a_{l,c}) \oplus 2.s(a_{2,c}) \oplus 3.s(a_{3,c}) \quad (2)$$
$$S'_{3,c} = 3.s(a_{0,c}) \oplus s(a_{l,c}) \oplus s(a_{2,c}) \oplus 2.s(a_{3,c}) \quad (for\ 0 \le c < 4)$$

Equation (2) illustrates how related calculations on $S(a_{i,j})$ might yield the values of $2 \cdot S(a_{i,j})$ in 2S-box and $3 \cdot S(a_{i,j})$ in 3S-box. Thus, by consulting the 256-byte S-box table and performing XOR operations, one can accomplish the intricate modifications of the round function. It is clear that the SubBytes, ShiftRows, and MixColumns transformations in the Round Function Module can be totally replaced by a single 256-byte table lookup operation in order to carry out the encryption. In particular, the accelerator replaces intricate computations of the multiplicative inverse, multiplication, and matrix multiplication over the finite field GF (2^8), with operations involving the lookup of a single 256-byte table.

Since the AES method is based on the mathematical foundation of the finite field GF (2^8), we can optimise computations and streamline our design by utilising the finite field's mathematical operations and procedures. The following is a description of the 2S-box calculation process in detail. By applying a shift and XOR to the matching value in the S-box, each value in the 2S-box is retrieved. In the following two scenarios, the particular operation is carried out.

Supposed $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ is the value from looking up the S-box. In the first case, if $b_7=0$, $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ is shifted to the left by one bit, and then the last bit is filled with 0, and finally the value of 2S-box is $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$. The final result in the second case is obtained by doing a bitwise XOR with {0x1b} after all the operations in the first case if b7=1. To save memory hardware costs, the values in this table are derived from associated calculations rather than being stored in the 3S-box. By bitwise XORing the matching values from the S-box and 2S-box, each value in the 3S-box is computed.

The calculation method of the values of 2S-box and 3S-box can be demonstrated by using mathematical calculations and procedures over the finite field GF (2^8). The suggested technique of searching through a single 256-byte table can be used to create a 128-bit AES accelerator, as demonstrated by the following mathematical proof. The following polynomial representation is used to represent all bytes as the finite field values in accordance with the AES algorithm specification:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i \quad (3)$$

The binary polynomial multiplication over the GF (2^8), however, can be used to demonstrate the above 2S-box calculation approach. In GF (2^8) multiplication with polynomials, the polynomials are multiplied modulo an irreducible polynomial of the form $m(x) = x^8 + x^4 + x^3 + x + 1$. Since the addition operation in GF (2^8) is the addition of modulo 2, the addition in the finite field can be represented by an XOR action.

In the binary polynomial representation, $x \cdot b(x)$ is used to represent {0x02} multiplied by the value b(x), where b(x) has coefficients $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ and each b_n is either 0 or 1. The calculation of $x \cdot b(x)$ is as follows (using $\cdot$ and $\oplus$ to represent finite field multiplication and addition (XOR), respectively):

$$x \cdot b(x) = x.(b_7x^7 \oplus b_6x^6 \oplus b_5x^5 \oplus b_4x^4 \oplus b_3x^3 \oplus b_2x^2 \oplus b_1x \oplus b_0)$$
$$= b_7x^8 \oplus b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus b_3x^4 \oplus b_2x^3 \oplus b_1x^2 \oplus b_0x \quad (4)$$

Then equation (4) is divided by the irreducible polynomial m(x) and takes the remainder as below:

$$= b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus b_3x^4 \oplus b_2x^3 \oplus b_1x^2 \oplus b_0x \oplus b_7.(x^4 \oplus x^3 \oplus x \oplus 1) \quad (5)$$

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

It is clear from equation (5) that the value in the 2S-box can be calculated in a proven way. The output of Equation (5) is the byte $\{b_6b_5b_4b_3b_2b_1b_{00}\}$ if b7 is equal to 0. Thus, the 2S-box calculation method's initial case is proven. When b7 equals 1, the polynomial of {00011011} (i.e., {0x1b}) can be used to determine the remainder. This can be done simply by shifting $\{b_7b_6b_5b_4b_3b_2b_1b_{00}\}$ to the left by one bit, filling the final one with 0, and then performing a bitwise XOR with {0x1b}. The 3S-box computation method can be shown by using the polynomial multiplication over the GF (2^8). The following computations are made using {0x03} multiplied by b(x) to represent any one value in the 3S-box and b(x) to represent any one value in the S-box:

$$\{03\}\bullet b(x) = (\{02\} \oplus \{01\}) \bullet b(x)$$

$$= (\{02\} \bullet b(x) \oplus \{01\} \bullet b(x)$$

$$= (\{02\} \bullet b(x)) \oplus b(x). \quad (6)$$

To interact with the hardware and perform encryption and decryption tasks, use the MATLAB software. Combining these technologies allows you to develop a dependable and efficient AES encryption system using hardware implementation and tried-and-true algorithms. In this proposed system in Table 1 the Encryption and decryption time taken based on the size of data hidden is shown below

Table 1: Comparative Analysis with Related Works

Title & Year	Algorithm	Data Size	Encryption time	Decryption time
VLSI Implementation of AES Block Cipher based Data Hiding (Proposed Work)	AES Algorithm	1024 Bytes/1Kb	1.15 ms	1.05 ms
Exploration on Accounting Data Encryption Processing System Based on DES Algorithm [1] (2023)	DES Algorithm	1024 bytes/1Kb	15.2 ms	12.6 ms
New Modified Twofish For Data Protection Using Salsa20 and Lu system [2] (2019)	Twofish Algorithm	10 KB	0.1188 sec	0.1186 sec
Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm [4] (2018)	RSA Algorithm	1024 bytes/1Kb	4.0 ms	29.0 ms

4. Results and Discussion

The design of the programme determines the output that a MATLAB programme implementing AES will produce. An encrypted and decrypted output is provided here as follows:

Encryption

Here, the programme uses AES to encrypt a plaintext message; the encrypted ciphertext in hexadecimal format may be the result. Binary data is frequently represented in a human-readable fashion using hexadecimal.

A detailed illustration of the several steps in the AES encryption process is given in Figures 5 through 10. The input image used for data concealing is shown in Figure 5, and the wavelet-transformed image used in the AES encryption process is shown in Figure 6. Figure 7 shows the process of creating the encryption input key. The Stego image that was produced following data embedding using AES encryption is shown in Figure 8. Figure 10 shows the successful password verification procedure when the correct input key is supplied, while Figure 9 illustrates the effect of an invalid password, which is caused by an unsuitable input key. The main phases and results of the AES encryption process are depicted in these diagrams taken together.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

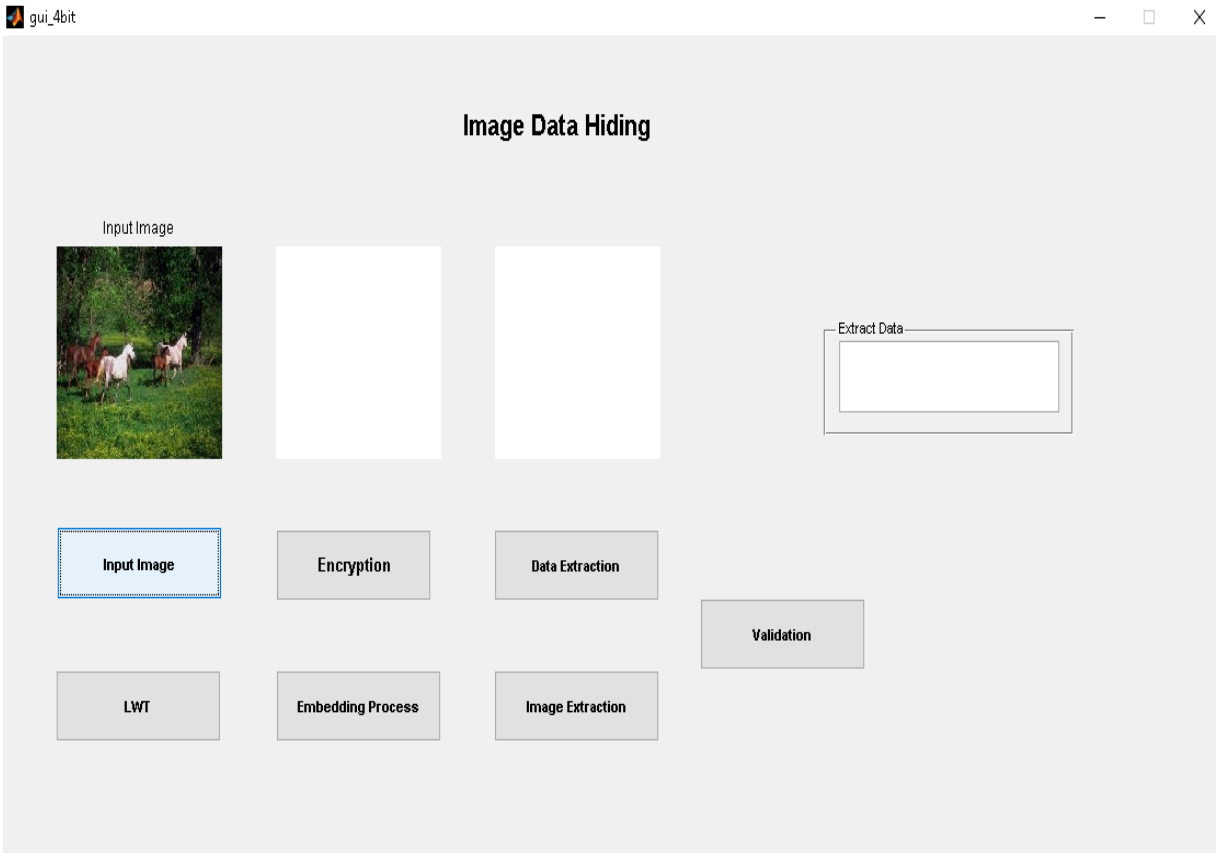

Figure 5: Input image

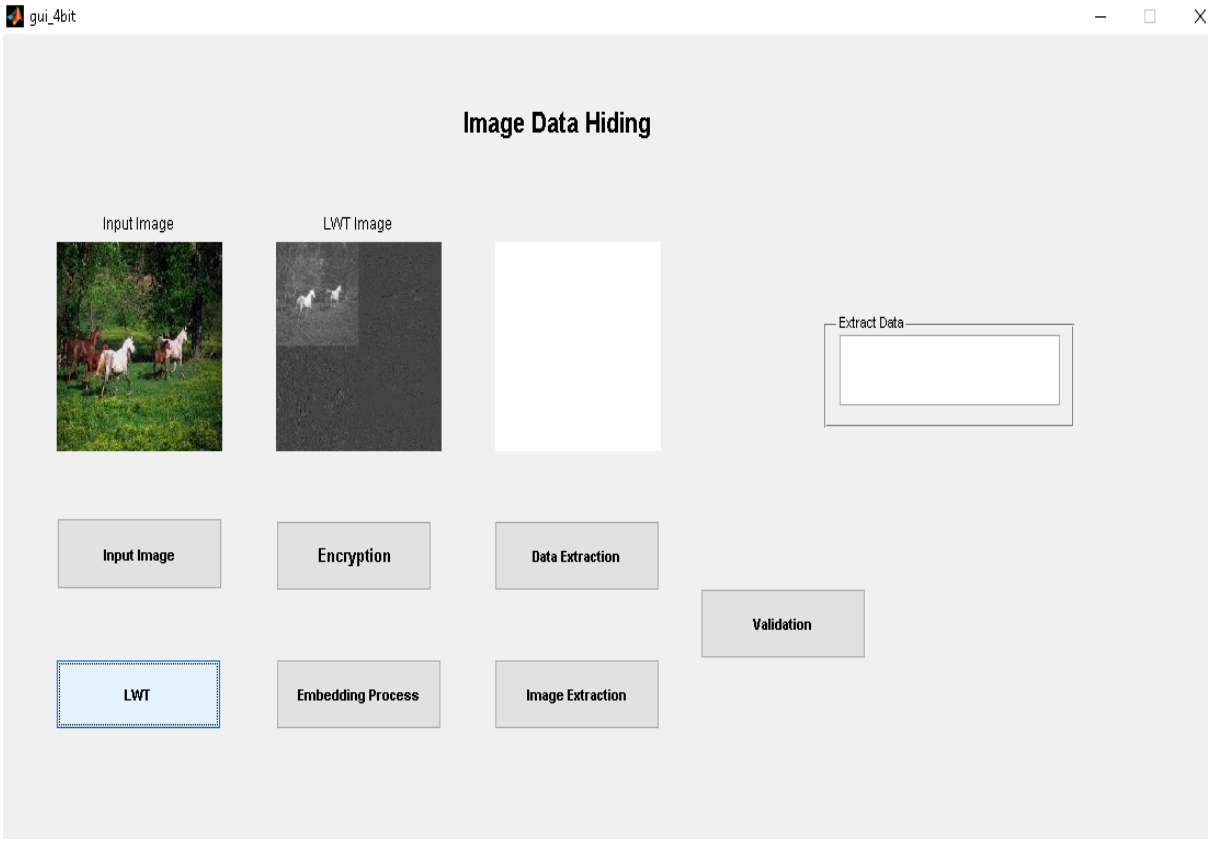

Figure 6: Wavelet transformed image

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

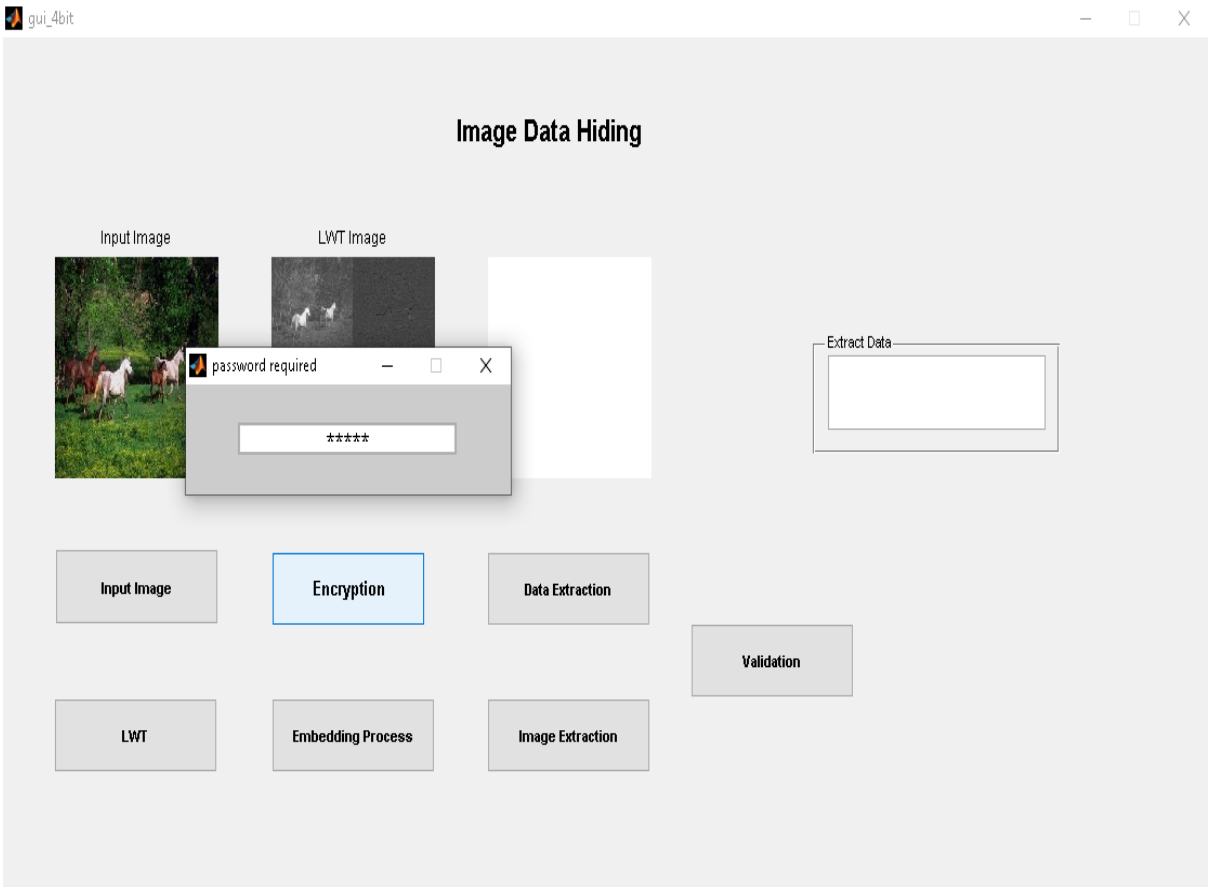

Figure 7: Input Key

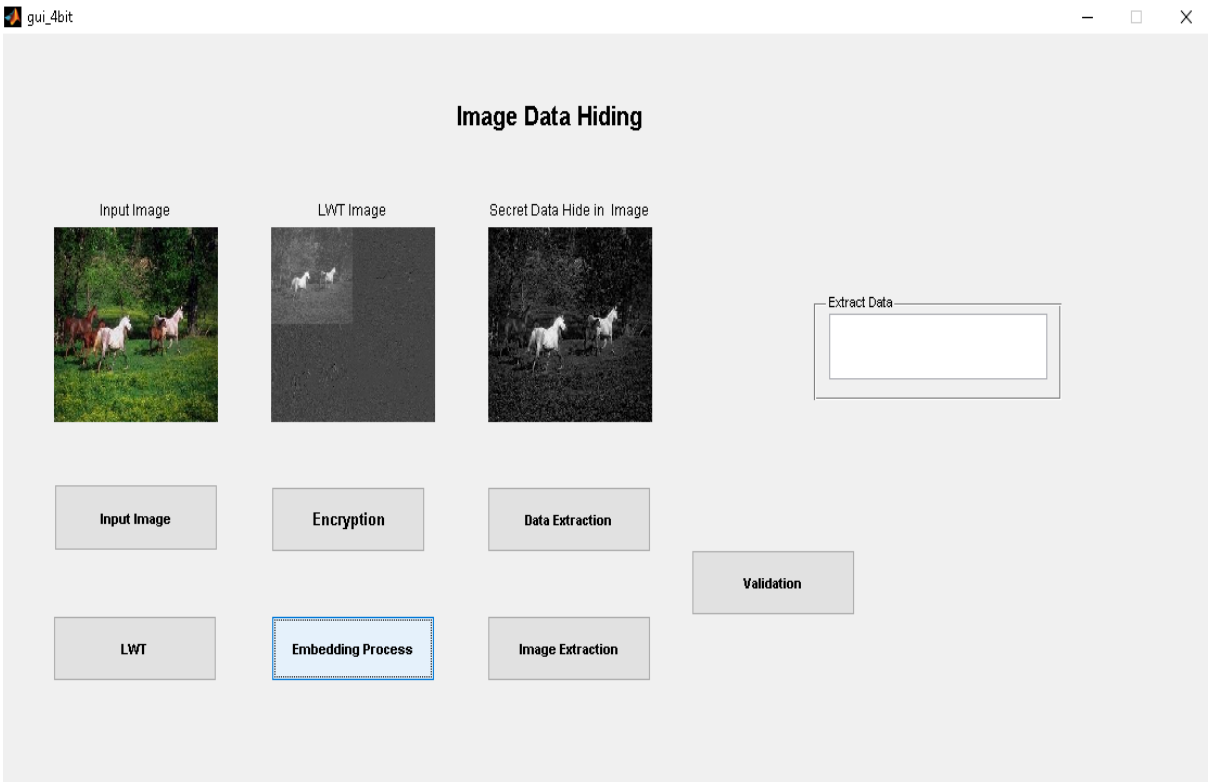

Figure 8: Stego Image

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

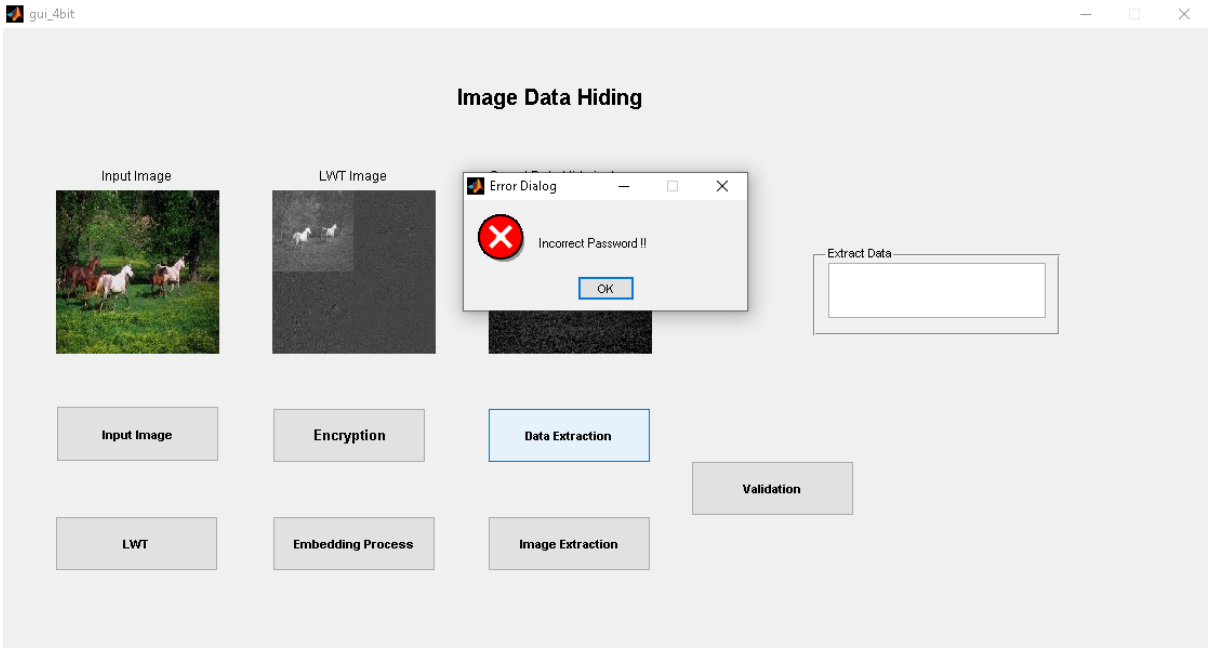

Figure 9: Incorrect password

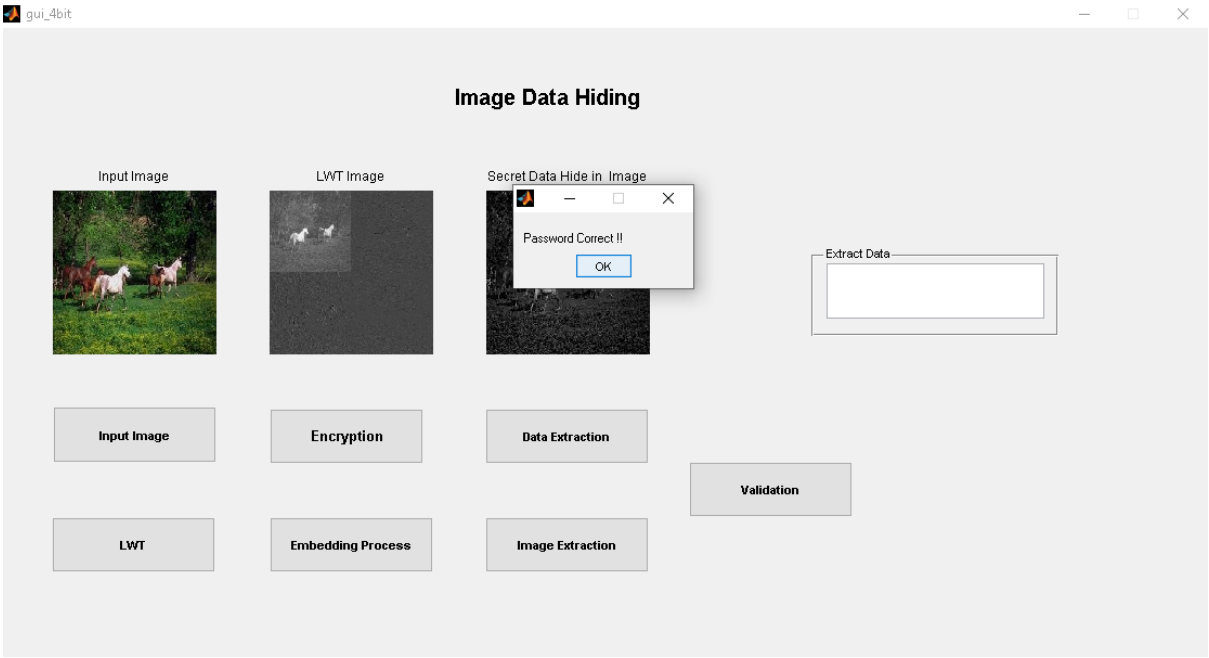

Figure 10: Password correct

Decryption

Here, the programme decrypts a ciphertext using AES and the relevant key; the output is the original plaintext message that was taken out of the ciphertext. The way your application is set up will determine whether this is shown as text or stored as a file.

The figures 11 to figures 14 demonstrate important results at different points in the AES decryption process. While Figure 12 shows the recovered image throughout the decryption process, Figure 11 emphasizes the retrieved data from the encrypted input. The recovered image is then shown in Figure 13 following the successful use of AES decryption. Lastly, Figure 14 verifies the accuracy and integrity of the decryption by validating the AES procedure. Together, these numbers give a comprehensive picture of the decryption process and its results.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

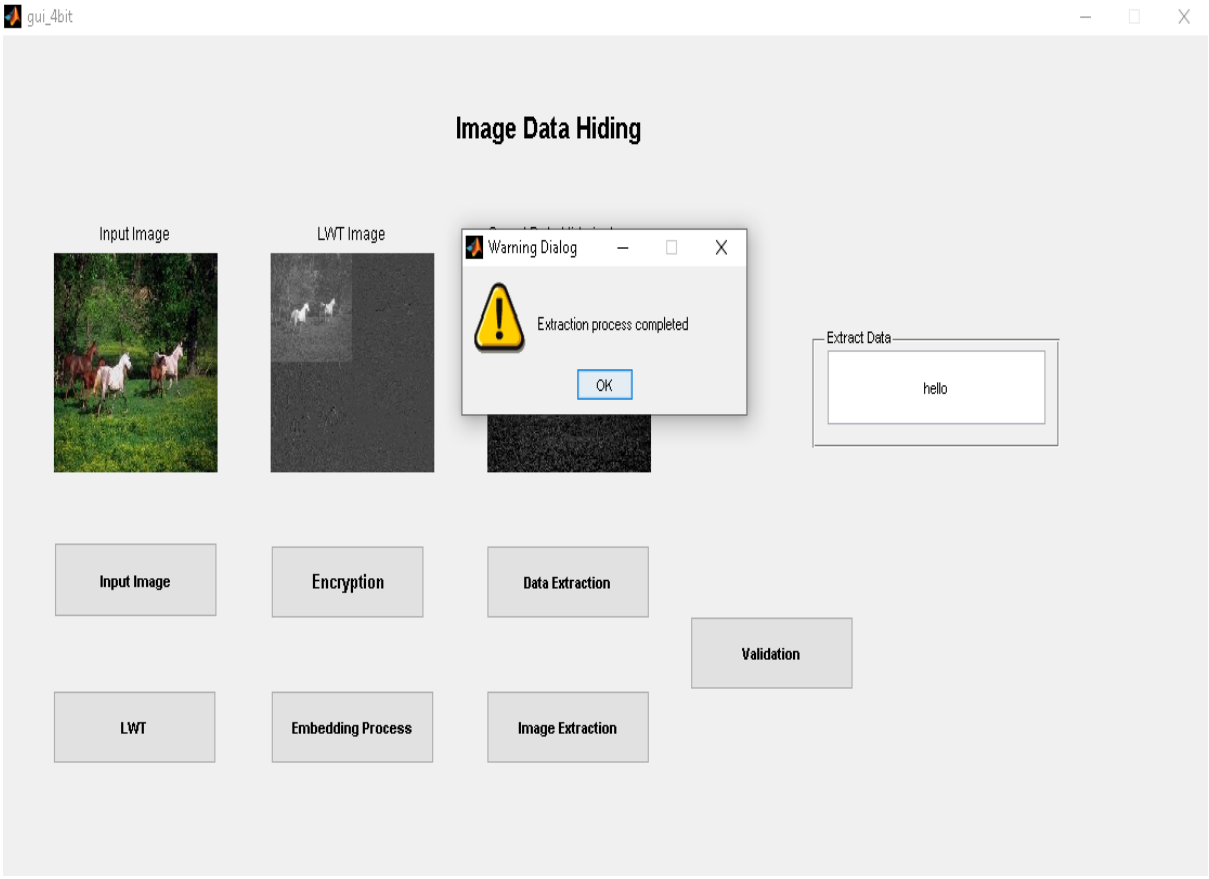

Figure 11: Extracted data

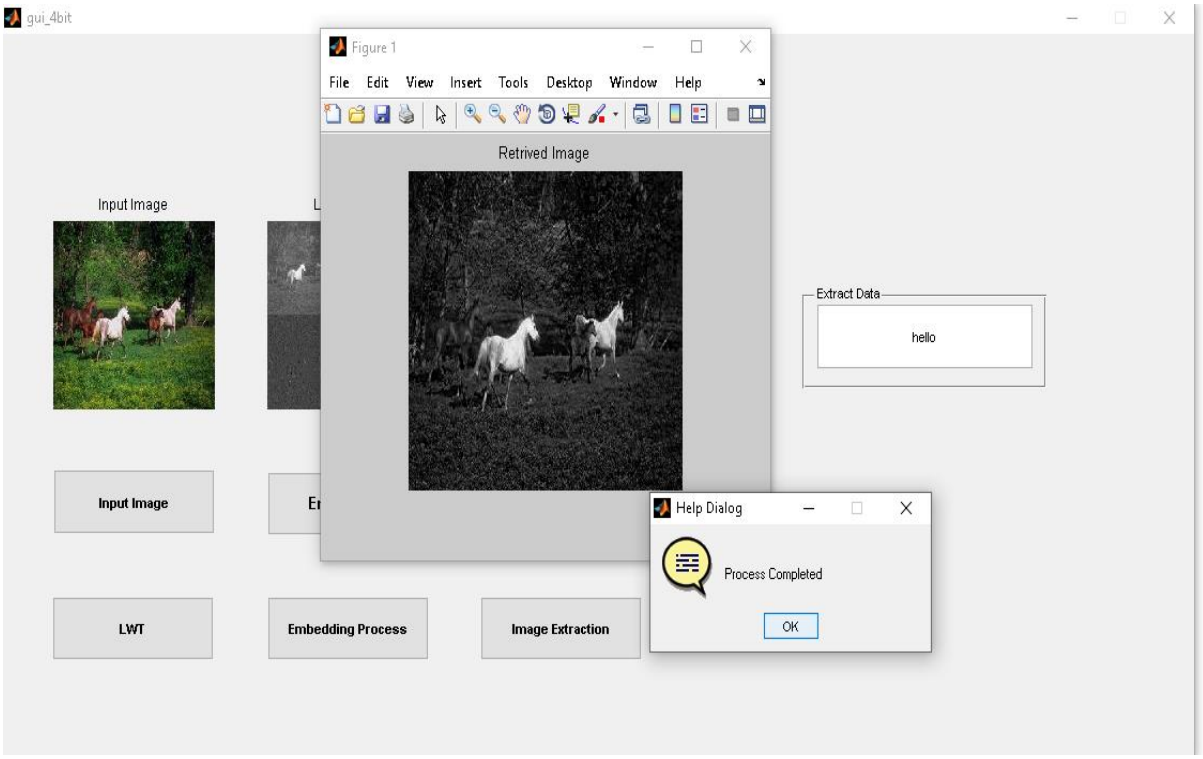

Figure 12: Image Recovered

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

Figure 13: Retrived Image

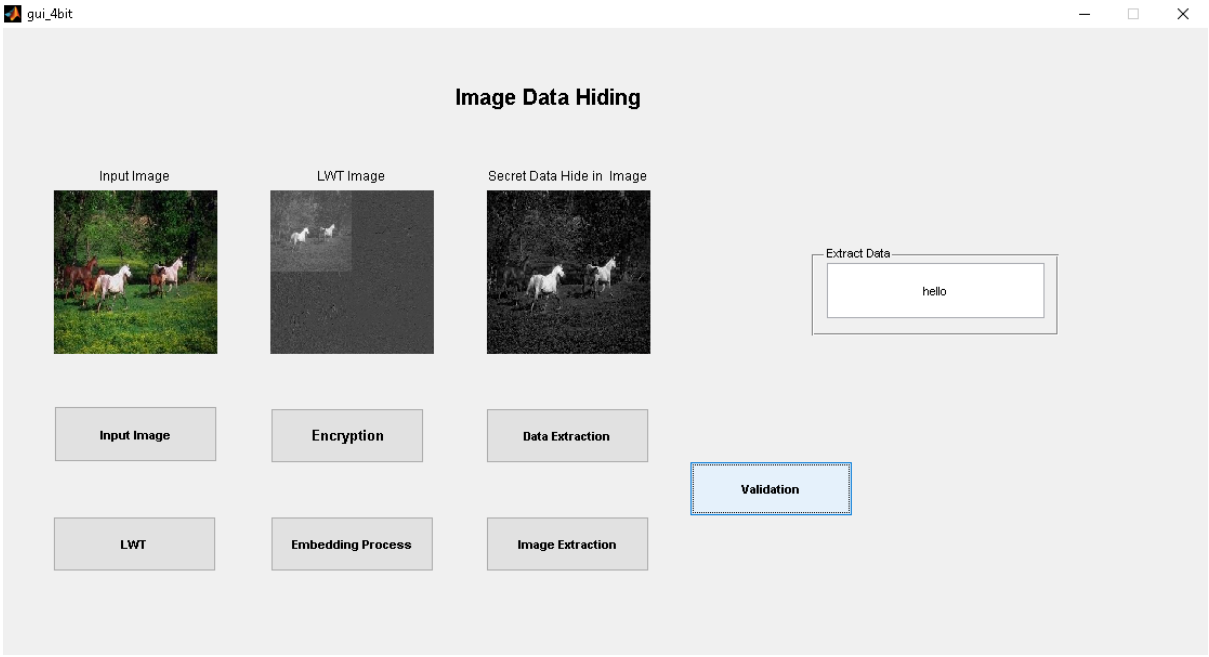

Figure 14: Validation of AES

Xilinx Output

Utilising Xilinx for hardware implementation and MATLAB for algorithm development, you may build a thorough analysis and perhaps implement an energy-efficient cooperative spectrum sensing architecture for cognitive radio networks.

The binary encoding of AES data, intended for VLSI (Very Large-Scale Integration) implementation, is shown in Figure 15. The process of transforming the encrypted or decrypted data into a binary sequence appropriate for hardware-level processing is shown in this image, guaranteeing effective integration and operation within VLSI systems.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

Figure 15: AES data in binary format for VLSI implementation

5. Conclusion and Future Scope

Implementing VLSI provides a strong method for protecting data that is concealed in pictures. This solution delivers a large security increase by encrypting the secret data with an AES block cypher before disguising it via LSB substitution and other techniques. In addition, compared to software-based solutions, the hardware acceleration offered by VLSI processors results in quicker processing, less power consumption, and possibly better data throughput. For real-time applications that need secure and effective data hiding in image processing, this combination is especially helpful. The complexity of hardware design, the possibility of image quality reduction as a result of data hiding, and the essential requirement for safe key management for both AES encryption and the data hiding procedure itself continue to be obstacles. In order to fully utilise this technology in practical situations, it will be imperative to address these issues.

The potential for investigating alternate data hiding strategies with greater resilience against picture manipulations is a promising aspect of the future scope of VLSI implementation for AES block cypher based data hiding in image processing. Research can also concentrate on reducing the effect on image quality while creating a VLSI architecture that is adaptable to different image formats and data-hiding capabilities. This may result in covert communication techniques that are safer and more effective.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] Aufa F. J., Endroyono and Affandi A., "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2018, pp. 1-5, doi: 10.1109/ICSTC.2018.8528584.

[2] El-Mowafy M. A., Gharghory S. M., Abo-Elsoud M. A., Obayya M. and Fath Allah M. I., "Chaos Based Encryption Technique for Compressed H264/AVC Videos," in IEEE Access, vol. 10, pp. 124002-124016, 2022, doi: 10.1109/ACCESS.2022.3223355.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025

[3] Hoomod H. K. and Hussein A. M., "New Modified Twofish For Data Protection Using Salsa20 and Lu system," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 1189-1195, doi: 10.1109/ICCS45141.2019.9065573.
[4] Kavuri V. R. and A. T. P., "Efficient Secured Cloud Storage System using Dynamic Multiple Clouds Cryptographic Algorithm," 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 2023, pp. 399-405, doi: 10.1109/I-SMAC58438.2023.10290155.
[5] Li.Z, "Exploration on Accounting Data Encryption Processing System Based on DES Algorithm," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-6, doi: 10.1109/AIKIIE60097.2023.10389923.
[6] Mahendra M. and Prabha P. S., "Classification of Security Levels to Enhance the Data Sharing Transmissions using Blowfish Algorithm In Comparison With Data Encryption Standard," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 1154-1160, doi: 10.1109/ICSCDS53736.2022.9760971.
[7] Parvathy P. and Remya Ajai A. S., "VLSI Implementation of Blowfish Algorithm for Secure Image Data Transmission," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0770-0774, doi: 10.1109/ICCSP48568.2020.9182088.
[8] Qingmin Y. and Sinan W., "Signature system based on RSA algorithm and its algorithm optimization," 2021 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), Shenyang, China, 2021, pp. 703-706, doi: 10.1109/TOCS53301.2021.9688864.
[9] Ramtri G. and Patel C., "Secure Banking Transactions Using RSA and Two Fish Algorithms," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.236.
[10] Shahbazi K. and Ko S. B., "Area-Efficient Nano-AES Implementation for Internet-of-Things Devices," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 136-148, Jan. 2021, doi: 10.1109/TVLSI.2020.3033928.
[11] Smekal D., Hajny J. and Martinasek Z., "Hardware-Accelerated Twofish Core for FPGA," 2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, Greece, 2018, pp. 1-5, doi:10.1109/TSP.2018.8441386.
[12] Vanitha M., Sakthivel R. and Subha, "Highly secured high throughput VLSI architecture for AES algorithm," 2012 International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2012, pp. 403-407, doi: 10.1109/ICDCSyst.2012.6188751.
[13] Venkata Rao D., Pavan Kumar Reddy S., Anusha C., Bhoomika D. and Venkateswarlu R., "Image Security is Improved by Super Encryption using RSA and Chaos Algorithms," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 1-5, doi: 10.1109/ICEARS56392.2023.10085142.
[14] ZhaoJ., "DES-Co-RSA: A Hybrid Encryption Algorithm Based on DES and RSA," 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 2023, pp. 846-850, doi: 10.1109/ICPECA56706.2023.10075771.
[15] S. Kumarganesh, S. Anthoniraj, T.Senthil Kumar, P.Elayaraja, et al. (2022), “A Novel Analytical Framework is Developed for Wireless Heterogeneous Networks for Video Streaming Applications” Journal of Mathematics, 2022(1) pp.1-7 doi:https://doi.org/10.1155/2022/2100883.
[16] Thiyaneswaran B, Anguraj K, Kumarganesh S and Thangaraj K, (2021), “Early Detection of Melanoma Images using gray level co-occurrence matrix Features and Machine Learning Techniques for Effective Clinical Diagnosis” International Journal of Imaging Systems and technology, Wiley Publications, Volume 31, Issue 2, Pages 682-694.

DOI: https://doi.org/10.54216/IJWAC.090101
Received: October 02, 2024 Revised: November 12, 2024 Accepted: January 07, 2025